International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

LANE DETECTION USING MACHINE LEARNING

[1]Mr M. CHINNA BABU, [2]CHINTHALA SANKEERTHANA , [3] AEROLLA SATHVIK RAO, [4] BONTHA VASANTH

[1](Assistant Professor) , CSE. Teegala Krishna Reddy Engineering College Hyderabad.

[2,3,4]B,tech , scholar , CSE. Teegala Krishna Reddy Engineering College Hyderabad.

ABSTRACT

Lane detection using OpenCV in Python is a crucial component of advanced driver assistance systems (ADAS) and autonomous vehicles, providing a foundation for enhancing road safety and facilitating automation in the transportation industry. The process involves several intricate steps, and these abstract aims to delve into the details of these procedures. The journey begins with preprocessing, where incoming video frames are initially converted to grayscale to simplify lane detection and minimize computational load. To further reduce noise and enhance edge detection, Gaussian blurring is applied. Subsequently, the Canny edge detection algorithm is employed to identify potential lane markings, highlighting the transitions in pixel intensity that correspond to edges in the image. After edge detection, a region of interest (ROI) is defined in the video frame to focus on the area of the road where lanes are typically located.

This step involves masking out irrelevant information and concentrating computational efforts on the most relevant region. Once the ROI is established, a perspective transformation is applied to create a bird's-eye view of the selected area. This transformation simplifies lane detection by rendering lane markings as straight lines, even if they curve in the original camera perspective. The Hough Line Transformation is then utilized to identify line segments in the bird's-eye view image that potentially correspond to lane markings. By parameterizing these lines, the system distinguishes between left and right lanes based on their slopes, allowing for the determination of the lane positions. To enhance system robustness and adaptability to changing road conditions, many lane detection implementations employ techniques such as exponential smoothing to

track the detected lanes over time. The final output is a video stream where the detected lane markings are overlaid onto the original frame, providing a visual representation of lane positions. This output serves various purposes, including assisting drivers in maintaining their lanes and contributing to the development of autonomous driving systems. Lane detection with OpenCV in Python exemplifies the potential of computer vision in modern transportation, enhancing safety and enabling the transition towards autonomous vehicles.

1. INTRODUCTION

Lane detection stands as a pivotal and indispensable component within the realm of modern advanced driver-assistance systems (ADAS) and autonomous vehicles. Its significance lies in its profound influence on enhancing road safety and facilitating the progression towards fully automated vehicle control. This introduction provides a foundational overview of the role that OpenCV, a robust computer vision library in the Python programming language, plays in the domain of detecting and tracking lane markings on roadways. In the world of automotive technology, the ability to accurately and robustly identify and follow lanes is paramount. It serves as the bedrock for numerous critical functions such as keeping vehicles within their designated lanes, enabling safe lane changes, and assisting drivers in maintaining optimal road positioning. Lane detection systems achieve these objectives through a combination of image processing, advanced computer vision techniques, and mathematical algorithms. These systems leverage video streams captured by cameras mounted on vehicles to provide realtime analysis of the road environment. OpenCV, as an open-source computer vision library, stands out as the go-to choice for this purpose, primarily due to its versatility and comprehensive set of tools designed for image analysis and manipulation.The primary objective of lane detection systems revolves around processing video frames, identifying lane markings, and continuously tracking them as the vehicle moves along the road. These steps culminate in the successful identification and tracking of the lanes, which are then overlaid onto the original video frames for visualization purposes.

These steps typically include grayscale conversion, edge detection to highlight lane markings, region of interest (ROI) selection to focus on the relevant part of the image, perspective transformation to rectify the

image's viewpoint, and the application of the Hough Line Transformation to identify and represent the lanes within the image. This is designed to delve into the intricate details of lane detection using OpenCV in the Python programming language. It will offer a comprehensive understanding of the key concepts involved in the process, detailed step-by-step implementation instructions, and insights into the significance of each stage in the lane detection pipeline. Whether you are a novice in the field of computer vision or an experienced developer looking to expand your knowledge, this guide will prove to be a valuable resource. By equipping with the knowledge and skills to implement this critical technology, it contributes to the broader objective of making vehicles safer and more capable of autonomous operation. Ultimately, the fusion of OpenCV and lane detection is a cornerstone in the journey towards safer and more autonomous road transportation.

2 . LITERATURE SURVEY

A literature review typically involves summarizing and analyzing existing research and publications related to lane detection using OpenCV and Python. An overview and some key references are:

1. **INTRODUCTION TO LANE DETECTION**:

Lane detection is a critical aspect of autonomous vehicles and driver assistance systems. It involves identifying and tracking lanes on the road to facilitate vehicle control and navigation.

2. **TRADITIONAL LANE DETECTION TECHNIQUES:**

HOUGH TRANSFORM:

The Hough transform is a technique in which features are extracted that is used in image analysis and digital image processing. Previously the classical Hough Transform worked on the identification of lines in the image but later it has been extended to identifying positions of shapes like circles and ellipses. In automated analysis of digital images, there was a problem of detecting simple geometric shapes such as straight lines, circle, etc

[1]. So in the pre-processing stage edge detector has been used to obtain points on the image that lie on the desired curve in image space. But due to some imperfections in image data or in the edge detector, some

pixels were missing on the desired curve as well as spacial deviation between the geometric shape used and the noisy edge pixels obtained by the edge detector. So to refine this problem Hough transform is used. In this the grouping of edge pixels into an object class is performed by choosing appropriate pixels from the set of parametric image objects

[2]. The simplest case of Hough transform is finding straight lines that are hidden in large amounts of image data. For detecting lines in images, the image is first converted into binary image using some form of thresholding and then the positive or suitable instances are added into the dataset. The main part of Hough transform is the Hough space. Each point (d, T) in Hough space matches to a line at angle T and distance d from the origin in the data space. The value of a function in Hough space gives the point density along a line in the data space.

EDGE DETECTION:

Edge detection works on the idea of the identification of points in the digital image at which the image brightness changes sharply. The points at which image brightness changes sharply are organized into a set of curved line segments termed as edges. Edge detection is a fundamental tool in image processing particularly in the areas of feature detection and extraction. Applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified. However, it is not always possible to obtain such ideal edges from real life images of moderate complexity. The Canny edge detector is an edge detection algorithm that uses a multiple stage algorithm so as to detect edges in images[3]. Its aim is to discover the optimal edge detection. In this definition, an optimal edge detector includes the following things • Good detection – the algorithm should be able to detect as many real edges in the image as possible. • Good localization – edges marked through this algorithm should approach as close as possible to the edge in

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

the real image. • Minimal response – a given edge in the image should only be marked once so as to reduce false edge

BILATERAL FILTER:

Bilateral filter is a simple and non-iterative scheme which smoothens the image while preserving the edges. The basic idea behind the working of bilateral filter is that the two pixels should be close to one another[4]. This filter split an image into large-scale features i.e. structure and small scale features i.e. texture. In this filter every sample is replaced by a weighted average of its neighbors. These weights reflects two forces i.e. the closeness of the neighborhood with the center sample so that larger weight is assigned to the closer samples, and similarity between neighborhood and the center sample so that larger weight is assigned to the similar samples[5].

3 . SYSTEM DESIGN

The system design for "lane detection using machine learning" involves architecting the components, algorithms, and workflow to create a cohesive and functional lane detection

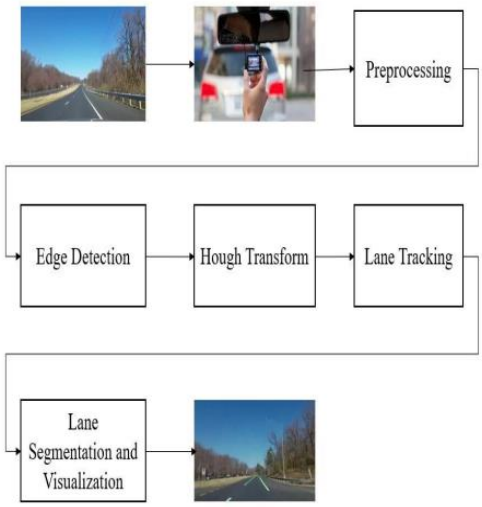


Fig-3.1 System Architecture

3.1 UML DIAGRAMS

3.1.1 CLASS DIAGRAM

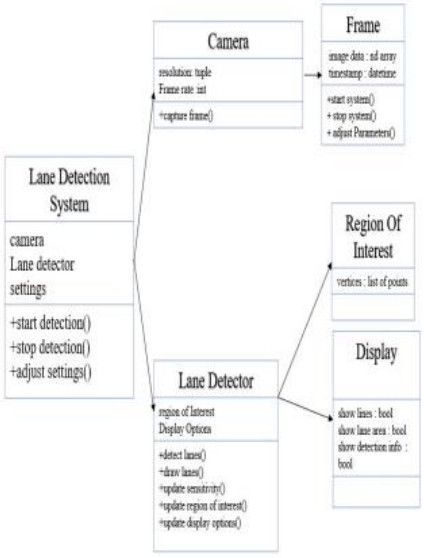


Fig-3.1.1 Class Diagram

The above picture depicts the class diagram of lane detection using machine learning.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

3.1.2 USECASE DIAGRAM

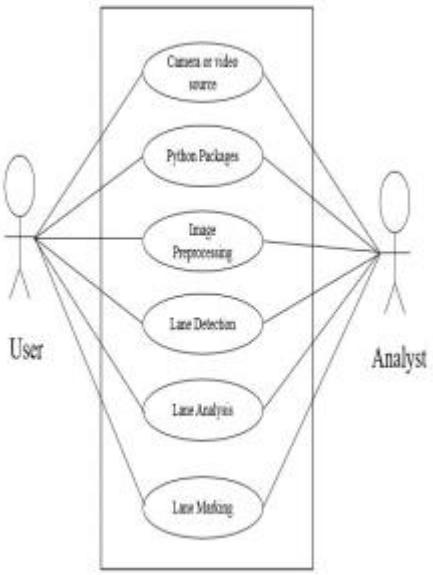


Fig-3.1.2 Use case Diagram

The above diagram describe the use cases required for lane detection using machine learning.

3.1.3 ACTIVITY DIAGRAM

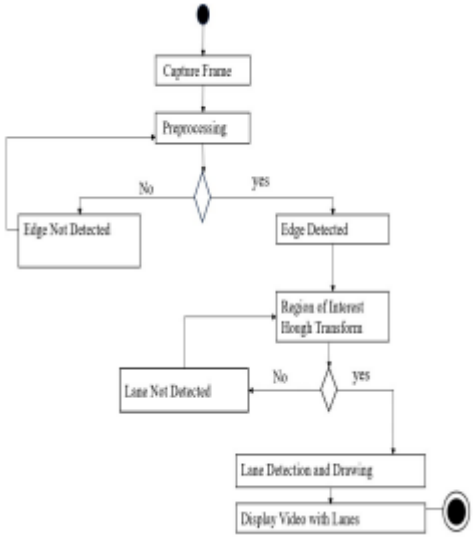


Fig-3.1.3 Activity Diagram

The above diagram describe the flow between the actions in an activity required for Lane Detection using Machine Learning.

3.1.4 STATECHART DIAGRAM

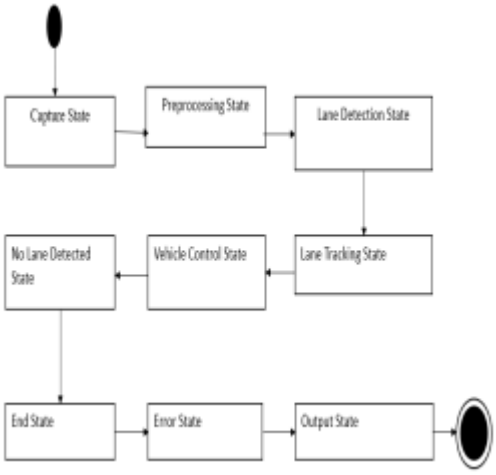
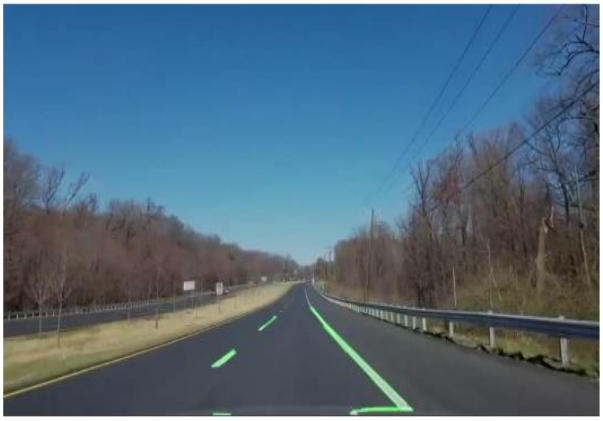


Fig-3.1.4 State chart Diagram

The above diagram describe the flow of control of an object required for Lane Detection using Machine Learning.

4 . OUTPUT SCREEN



. Fig-4.1 Original Frame

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

The output of a lane detection system using machine learning and OpenCV is a visual representation of the original frame with accurately identified and marked lane lines, helping to enhance road awareness and navigation

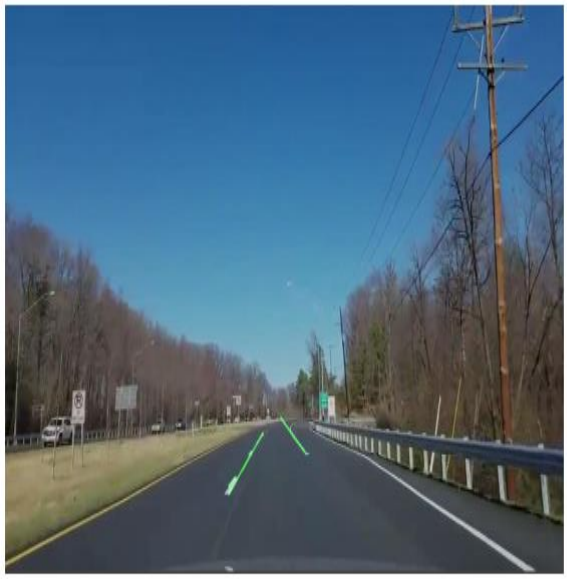


Fig-4.2 Output Screen

5 . CONCLUSION

In conclusion, lane detection in OpenCV using Python is a crucial component of advanced driver assistance systems and autonomous vehicles. It plays a pivotal role in ensuring vehicle safety and control by providing real-time information about lane boundaries. OpenCV, with its extensive computer vision capabilities, offers a solid foundation for developing robust and accurate lane detection algorithms. The implementation of lane detection can be greatly improved by incorporating advanced techniques such as semantic segmentation, dynamic region-of-interest selection, and post-processing methods. These enhancements can enhance the accuracy and adaptability of the system, making it more reliable in various road and lighting conditions. The development and continuous refinement of lane detection algorithms in OpenCV will continue to play a pivotal role in advancing autonomous driving technology and making our roadways safer for all. As technology continues to evolve, the integration of sensor fusion and machine learning approaches can further improve the precision of lane detection in real-world scenarios. Lane detection in OpenCV, as a fundamental building block of autonomous driving systems, will continue to be an active area of research and development, contributing to the advancement of the automotive industry and enhancing road safety in the years to come.

6 . FUTURE ENHANCEMENT

Enhancing lane detection in OpenCV using Python involves improving the robustness and accuracy of the existing algorithm. There are several techniques you can employ to achieve this. First, consider

incorporating advanced computer vision techniques such as semantic segmentation to differentiate between lane markings and other road elements. This can help reduce false positives and improve the accuracy of lane detection, especially in challenging lighting and road conditions. Additionally, using deep learning models like Convolutional Neural Networks (CNNs) for feature extraction can enhance lane detection by learning more complex patterns and handling diverse road scenarios. Second, implement dynamic region-of-interest (ROI) selection to adapt to changing road conditions and vehicle positions. This can be achieved by using sensor data like GPS and IMU information to adjust the region of interest in real-time. By dynamically updating the ROI, you can focus on the relevant road area, improving the system's adaptability to different driving scenarios and environmental changes. Third, consider post-processing techniques to smooth the detected lane lines and reduce jitter. Applying techniques like Kalman filtering or polynomial curve fitting can help create stable and accurate lane representations. Additionally, introducing lane tracking algorithms can make the system more robust, allowing it to handle temporary occlusions and lane changes more effectively. These enhancements can collectively improve the performance and reliability of your lane detection system in OpenCV with Python.

7 . REFERENCES

[1]-B. Fardi, U. Scheunert, H. Cramer, and G. Wanielik. A new approach for lane departure identification. Symposium, 2003. Proceedings. IEEE, pages 100--105, June 2003.

[2]-Thorpe C., Hebert M., Kanade T., Shafer S.: Toward autonomous driving: the CMU Navlab. Part I: perception. IEEE Expert 6, 31–42 (1991)

[3]- Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing, 22(10):761--767, 2004

[4]-Zu Kim. Realtime lane tracking of curved local road. In Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE, pages 1149--1155. IEEE, 2006.

[5]-Mohamed Aly. Real time detection of lane markers in urban streets. In Intelligent Vehicles Symposium, 2008 IEEE, pages 7--12. IEEE, 2008

[6]-D. Schreiber, B. Alefs, and M. Clabian. Single camera lane detection and tracking. In Intelligent Transportation Systems, 2005. pages 302--307,Sept 2005

[7]- Joel C McCall and Mohan M Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. Intelligent Transportation Systems, IEEE Transactions on, 7(1):20--37, 2006.

[8] S. Sivaraman and M.M. Trivedi. Integrated lane and vehicle detection, and trackingIEEE Transactions on Intelligent Transportation Systems, 14(2):906--917, June 2013

[9] B. Fardi, U. Scheunert, H. Cramer, and G. Wanielik. lane departure identification. In Intelligent Vehicles Symposium, 2003. Proceedings. IEEE, pages 100--105, June 2003

[10] Amol Borkar and Mark T Smith. Robust lane detection and tracking with ransac and kalman filter. 2009 16th IEEE International Conferenceon, pages 3261--3264. IEEE, 2009

[11] Mohamed Aly. Real time detection of lane markers in urban streets. In Intelligent Vehicles Symposium, 2008 IEEE, pages 7--12. IEEE, 2008.